IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1-22. Cancelled.

23. (Currently Amended) A method for a scheduled execution of a target function by a processor of a computer at predetermined times, wherein the processor comprises a first interrupt input operable to receive a first interrupt signal, the method comprising:

executing a start function, wherein the start function is executed by the processor as a first interrupt service routine, wherein the start function is executed in response to triggering of the first interrupt signal;

the start function repeatedly reading a computer register to obtain a read value;

the start function comparing the read value with a reference value, wherein the reference value corresponds to a predetermined time; and

the start function calling ~~executing~~ the target function ~~in the processor~~ when the read value corresponds to the reference value, wherein the start function calling the target function ~~is operable to~~ initiates said execut[[ing]]ion of the target function.

24. (Previously Presented) The method of claim 23, wherein the first interrupt signal is triggered by a timer, wherein the timer is programmed with the reference value by the start function.

25. (Currently Amended) The method of claim 23,

wherein the first interrupt signal is triggered with a lead time before the predetermined time;

wherein the lead time is ~~substantially~~ greater than an expected maximum delay between appearance of the first interrupt signal at the first interrupt input of the processor and said execut[[ing]]ion of the start function.

26. (Previously Presented) The method of claim 23, wherein the computer register comprises a count register.

27. (Previously Presented) The method of claim 26, wherein the count register comprises a time stamp counter of the processor.

28. (Previously Presented) The method of claim 23, wherein the first interrupt signal is triggered by a computer timer as a timer interrupt.

29. (Previously Presented) The method of claim 23, wherein an expected maximum delay is determined continuously.

30. (Currently Amended) The method of claim 29,

wherein the computer register comprises a count register; and

wherein the value for the expected maximum delay is determined by using an actual delay, wherein the actual delay is determined by reading the count register at the beginning of the start function and by subtracting the value representing time of appearance of the first interrupt signal.

31. (Previously Presented) The method of claim 30, wherein the expected maximum delay is determined by multiplying the actual delay by a safety factor.

32. (Previously Presented) The method of claim 31, further comprising:

setting value of a lead time substantially equal to the expected maximum delay when the expected maximum delay exceeds an upper limit.

33. (Previously Presented) The method of claim 30, further comprising:

generating an error when the expected maximum delay exceeds an upper limit.

34. (Previously Presented) The method of claim 29, wherein a timer interrupt is operable to be used by other programs running simultaneously on the computer to call an original function.

35. (Previously Presented) The method of claim 34, further comprising:

reading an address of the original function from an interrupt table, wherein the interrupt table contains one or more addresses of service routines associated with one or more interrupt inputs; and

replacing the address of the original function by the address of the start function in the interrupt table.

36. (Currently Amended) The method of claim 34, wherein one or more of the ~~scheduled~~ target function and the original function are operable to be executed by an interrupt request by means of the start function.

37. (Previously Presented) The method of claim 34, wherein the timer is operable to be adjusted to a clock rate by the operating system, and wherein the timer is operable to be set to a maximum clock rate prior to said repeatedly reading.

38. (Previously Presented) The method of claim 37, wherein the clock rate of the timer is changed and reset to the maximum clock rate prior to said executing the start function.

39. (Previously Presented) The method of claim 34, further comprising:

upon triggering the first interrupt signal, creating one or more of a list of one or more predetermined times for the execution of the target function and a list of one or more times for triggering of the first interrupt signal;

wherein the start function compares a next time of the execution of the target function with a time of a next interrupt signal and operates to cause execution of the original function if the next interrupt signal appears at least a maximum delay before a next time of execution of the target function.

40. (Currently Amended) The method of claim 34, further comprising:

upon triggering the first interrupt signal, creating one or more of a list of one or more predetermined times for the execution of the target function and a list of one or more times for execution of the original function[[.]];

wherein the start function compares a next time of the execution of the target function with a time of a next time of execution of the original function and operates to cause execution of the original function if the next time of execution of the original function appears at least a maximum delay before a next time of execution of the target function.

41. (Previously Presented) The method of claim 23, further comprising:

changing value of the register by the start function;

wherein said changing value of the register comprises pushing the register contents onto a computer stack at a beginning of the start function, and wherein the value of the register are written back into the register at an end of the start function.

42. (Previously Presented) The method of claim 23, further comprising:

determining a currently executed interrupt by reading a register of an interrupt controller at a beginning of the start function; and

acknowledging processing of the current interrupt request by an end-of-interrupt command (EOI) after said determining the currently executed interrupt.

43. (Currently Amended) The method of claim [[23]]34, further comprising:

activating the original function, wherein said activating the original function comprises activating the original function by using a jump command by means of the start function.

44. (Previously Presented) The method of claim 23, further comprising:

the start function determining existence of an additional interrupt signal at the first interrupt input;

wherein the start function is operable to send an error to the target function if said determining the existence of the additional interrupt signal is true.

45. (Currently Amended) The method of claim 23, further comprising:

loading a software program into memory of the computer, wherein the software program is ~~operable to~~ executable to perform said execut[[e]]ing the start function, said repeatedly reading the computer register, said compare[[e]]ing the read value with a reference value, and said calling ~~executing~~ the target function ~~in the processor~~.

46. (Currently Amended) A [[M]]machine-readable data carrier with a software program stored on the data carrier, wherein the software program implements a method for the scheduled execution of a target function[[the,]] the software program ~~comprising the steps to~~ being executable by a processor to:

execute a start function, wherein the start function is executed by the processor as an interrupt service routine;

repeatedly read a computer register to obtain a read value, wherein said repeatedly reading comprises the start function repeatedly reading the computer register;

compare the read value with a reference value, wherein the reference value corresponds to a predetermined time; and

execute the target function in the processor when the read value corresponds to the reference value, wherein said executing is performed in response to the start function calling the target function when the read value corresponds to the reference value.

47. (Previously Presented) The machine-readable data carrier of claim 46, wherein said comparing the read value comprises the start function repeatedly comparing the read value.

48. (Previously Presented) The machine-readable data carrier of claim 46, wherein the interrupt signal is triggered by a timer, wherein the timer is programmed with the reference value by the start function.

49. (Currently Amended) The machine-readable data carrier of claim 46,

wherein the interrupt signal is triggered with a lead time before the predetermined time;

wherein the lead time is ~~substantially~~ greater than an expected maximum delay between appearance of the interrupt signal at the interrupt input of the processor and said executing the start function.

50. (Previously Presented) The machine-readable data carrier of claim 46, wherein the computer register comprises a count register.

51. (Previously Presented) The machine-readable data carrier of claim 50, wherein the count register comprises a time stamp counter of the processor.

52. (Previously Presented) The machine-readable data carrier of claim 46, wherein the interrupt signal is triggered by a computer timer as a timer interrupt.

53. (Previously Presented) The machine-readable data carrier of claim 46, wherein an expected maximum delay is determined continuously.

54. (Currently Amended) The machine-readable data carrier of claim 53,

wherein the computer register comprises a count register; and

wherein the value for the expected maximum delay is determined by using an actual delay, wherein the actual delay is determined by reading the count register at the beginning of the start function and by subtracting the value representing time of appearance of the interrupt signal.

55. (Previously Presented) The machine-readable data carrier of claim 54, wherein the expected maximum delay is determined by multiplying the actual delay by a safety factor.

56. (Previously Presented) The machine-readable data carrier of claim 55, wherein the software program further comprises the steps to:

set value of a lead time substantially equal to the expected maximum delay when the expected maximum delay exceeds an upper limit.

57. (Previously Presented) The machine-readable data carrier of claim 54, wherein the software program further comprises the steps to:

generate an error report when the expected maximum delay exceeds an upper limit.

58. (Currently Amended) The ~~method~~ machine-readable data carrier of claim 53, wherein a timer interrupt is operable to be used by other programs running simultaneously on the computer to call an original function.

59. (Currently Amended) The ~~method~~ machine-readable data carrier of claim 58, wherein the software program ~~further comprises the steps~~ is further executable by the processor to:

read an address of the original function from an interrupt table, wherein the interrupt table contains one or more addresses of service routines associated with various interrupt inputs; and

replace the address of the original function by the address of the start function in the interrupt table.

60. (Currently Amended) The ~~method~~ machine-readable data carrier of claim 58, wherein one or more of the ~~scheduled~~ target function and the original function are operable to be executed by an interrupt request by means of the start function.

61. (Previously Presented) The machine-readable data carrier of claim 58, wherein the timer is operable to be adjusted to a clock rate by the operating system, and wherein the timer is operable to be set to a maximum clock rate prior to said repeatedly reading.

62. (Previously Presented) The machine-readable data carrier of claim 61, wherein the clock rate of the timer is changed and reset to the maximum clock rate prior to said executing the start function.

63. (Previously Presented) The machine-readable data carrier of claim 58, wherein the software program ~~further comprises the steps~~ is further executable by the processor to:

upon triggering the interrupt signal, create one or more of a list of one or more predetermined times for the execution of the target function and a list of one or more times for triggering of the interrupt signal;

wherein the start function compares a next time of the execution of the target function with a time of a next interrupt signal and operates to cause execution of the original function if the next interrupt signal appears at least a maximum delay before a next time of execution of the target function.

64. (Previously Presented) The machine-readable data carrier of claim 46, wherein the software program further comprises the steps to:

change value of the register by the start function;

wherein said changing value of the register comprises pushing the register contents onto a computer stack at a beginning of the start function, and wherein the value of the register are written back into the register at an end of the start function.

65. (Previously Presented) The machine-readable data carrier of claim 46, wherein the software program further comprises the steps to:

determine a currently executed interrupt by reading a register of an interrupt controller at a beginning of the start function; and

acknowledge processing of the current interrupt request by an end-of-interrupt command (EOI) after said determining the currently executed interrupt.

66. (Currently Amended) The machine-readable data carrier of claim [[46]]58, wherein the software program further comprises the steps to:

activate the original function, wherein said activating the original function comprises activating the original function by using a jump command by means of the start function.

67. (Previously Presented) The machine-readable data carrier of claim 46, wherein the software program further comprises the steps to:

determine existence of an additional interrupt signal at the interrupt input by the start function;

wherein the start function is operable to send an error report to the target function if said determining the existence of the additional interrupt signal is true.